```
term_subst2_addprime  THM

⊢  ∀x:Term, as:(Label×Term) List.
|  (∀v:Label. (v ∈ term_vars(x)) ⇒ (v ∈ 1of(unzip(as))))
|  ⇒
|  term_subst2(as;(x)') = term_subst(as;x)
| by D 0 THENA Auto

|    THEN Unfolds ``term_subst term_subst2 addprime`` 0

|    THEN TermInd 1
THEN Reduce 0
THEN Try (Complete Auto)

|    THEN All (Folds ``term_subst term_subst2 addprime``)

|    THEN Try (Auto THEN EqCD THENA Auto

|              THEN BackThruSomeHyp THEN

|              RWW "member_append" (-1) THEN Auto

|              THEN BackThruSomeHyp THEN TrivialOr)
|
1. Term
2. u : Term→Type
3. x:{v:Term| u(v) }→∀as:(Label×Term) List.
|  (∀v:Label. (v ∈ term_vars(x)) ⇒ (v ∈ 1of(unzip(as))))
|  ⇒
|  term_subst2(as;(x)') = term_subst(as;x)
4. x2 : Label
⊢  ∀as:(Label×Term) List.
|  (∀v:Label. (v ∈ [x2]) ⇒ (v ∈ 1of(unzip(as))))
|  ⇒
|  apply_alist(as;x2;x2') = apply_alist(as;x2;x2)  by Auto
|
1. Term
2. u : Term→Type
3. x:{v:Term| u(v) }→∀as:(Label×Term) List.
   (∀v:Label. (v ∈ term_vars(x)) ⇒ (v ∈ 1of(unzip(as))))
   ⇒
   term_subst2(as;(x)') = term_subst(as;x)
4. x2 : Label
5. as : (Label×Term) List
6. ∀v:Label. (v ∈ [x2]) ⇒ (v ∈ 1of(unzip(as)))
⊢  apply_alist(as;x2;x2') = apply_alist(as;x2;x2)
  by Unfold `apply_alist` 0 THEN Unfold `find` 0

     THEN ListInd (-2) THEN Reduce 0

     THEN Try (Unfold `unzip` 0 THEN Reduce 0 THEN Auto

              THEN InstHyp [x2] (-1) THEN Auto

              THEN Try (RWW "member_singleton" 0 THEN Auto)
```

```
  ...
      )

  THEN Try (Reduce 0 THEN SplitOnConclITE THENA Auto

           THEN Reduce 0
      THEN Try BackThruSomeHyp

           THEN Auto THEN
      InstHyp [v@0] (-7)

           THENM RWW "cons_member" (-1)
      THENM D (-1)

           THENA Auto

           THEN Try (D (-4) THEN RWW "member_singleton" (-2)

                      THENM BackThruLemma ‘eq_lbl_iff‘

                      THENM RelRST
                THENA Auto

                )

           THEN Try (Unfold ‘unzip‘ 0 THEN Reduce 0 THEN Trivial)

      )
END
```